KJ-GRIPPER

USER MANUAL

K-Team

MOBILE ROBOTICS

VERSION 1.0
JUNE 2010

Documentation Author

Frederic Lambercy
K-Team S.A.
Rue Galilee 9, Y-Park
1400 Yverdon-les-Bains
Switzerland

Email: info@k-team.com
Url: www.k-team.com

Trademark Acknowledgements:

IBM PC: International Business Machines Corp.
Macintosh: Apple Corp.
SUN Sparc-Station: SUN Microsystems Corp.
LabVIEW: National Instruments Corp.
Matlab: MathWorks Corp.
K-Junior: K-Team.

LEGAL NOTICE:

- The contents of this manual are subject to change without notice
- All efforts have been made to ensure the accuracy of the content of this manual. However, should any error be detected, please inform K-Team.
- The above notwithstanding, K-Team can assume no responsibility for any error in this manual.

TABLE OF CONTENTS

1 INTRODUCTION

The hardware of K-Junior is based on a modular concept. The Gripper is a turret that can be plugged on the basic configuration making object manipulation possible. Due to the configuration of this turret, other turrets can be plugged on the top of it, but can't be under the Gripper. The gripper must be the first turret plugged on the K-Junior in order to keep mechanical fixings accessible.

In order to match which each configuration, the K-Junior Gripper is available in two different versions.

The K-Junior gripper has an additional Battery (1350mAh) to keep the autonomy of the K-Junior unchanged when used with the Gripper (up to 3h). The charger is on board, an external power supply is provided with the Gripper. The monitoring of the battery is made by the CPU.

A microcontroller installed in the gripper manages local functionalities: position and speed control of the arm and gripper, battery management and communication with the K-Junior. A I2C network connects this local microcontroller to the K-Junior processor. Some registers accessible by the I2C bus allows to control the Gripper from the main processor, where the high level control structure can be implemented.

1.1 How to use this handbook

This manual is organised into four chapters and an appendix. To learn how to make the best use of your gripper turret you are urged to read all of chapters 2 to 4. The appendix can be referred to as necessary.

- **Unpacking and Inspection:** K-Junior Gripper's package description and first start-up
- **The K-Junior Gripper:** Description of all the gripper's functionalities
- **Connections:** Explanation on how to connect (or disconnect) the gripper to the robot
- **Gripper use:** Instructions to program the gripper using the high level function or access directly to the gripper registers.

1.2 Safety precautions

Here are some recommendations on how to correctly use the K-Junior Robot:

- **Don't force mechanical movements of the gripper!** Movements of the gripper mechanical parts have to be controlled ONLY by software. Forcing arm or gripper (open or close) movements can damage the gripper mechanics!
- **Keep out from the wet places!** A contact with water can made a short circuit and damage the electronics.
- **Use only the official charger or the cable which is delivered with the Gripper.** Do not try to use another charger; this can cause irreversible damage to the battery.
- **Don't plug or unplug any connector or turret when the robot is powered!** All connections and turret insertions must be made when the robot and the interface are switched OFF. Otherwise damages can occur.
- **Never leave the K-Junior powered when it is unused**. When you have finished working with K-Junior, turn it off. It will save the battery life

1.3 Unpacking and inspection

Please check that you have a complete package. You should find:

- Documentation
- The Gripper Turret with a Battery pack
- External power supply to charge the onboard battery
- Two screws to fix the Gripper on the K-Junior

1.4 Recycling

Think about the end of life of your robot! Parts of the robot can be recycled and it is important to do so. It is for instance important to keep batteries out of the solid waste stream. When you throw away a battery, it eventually ends up in a landfill or municipal incinerator. These batteries, which contain Lithium Polymer, can contribute to the toxicity levels of landfills or incinerator ash. By recycling the batteries through recycling programs, you can help to create a cleaner and safer environment for generations to come. For those reasons please take care to the recycling of your robot at the end of its life cycle, for instance sending back the robot to the manufacturer or to your local dealer.

Thanks for your contribution to a cleaner environment!

2 The K-Junior Gripper

2.1 Overview

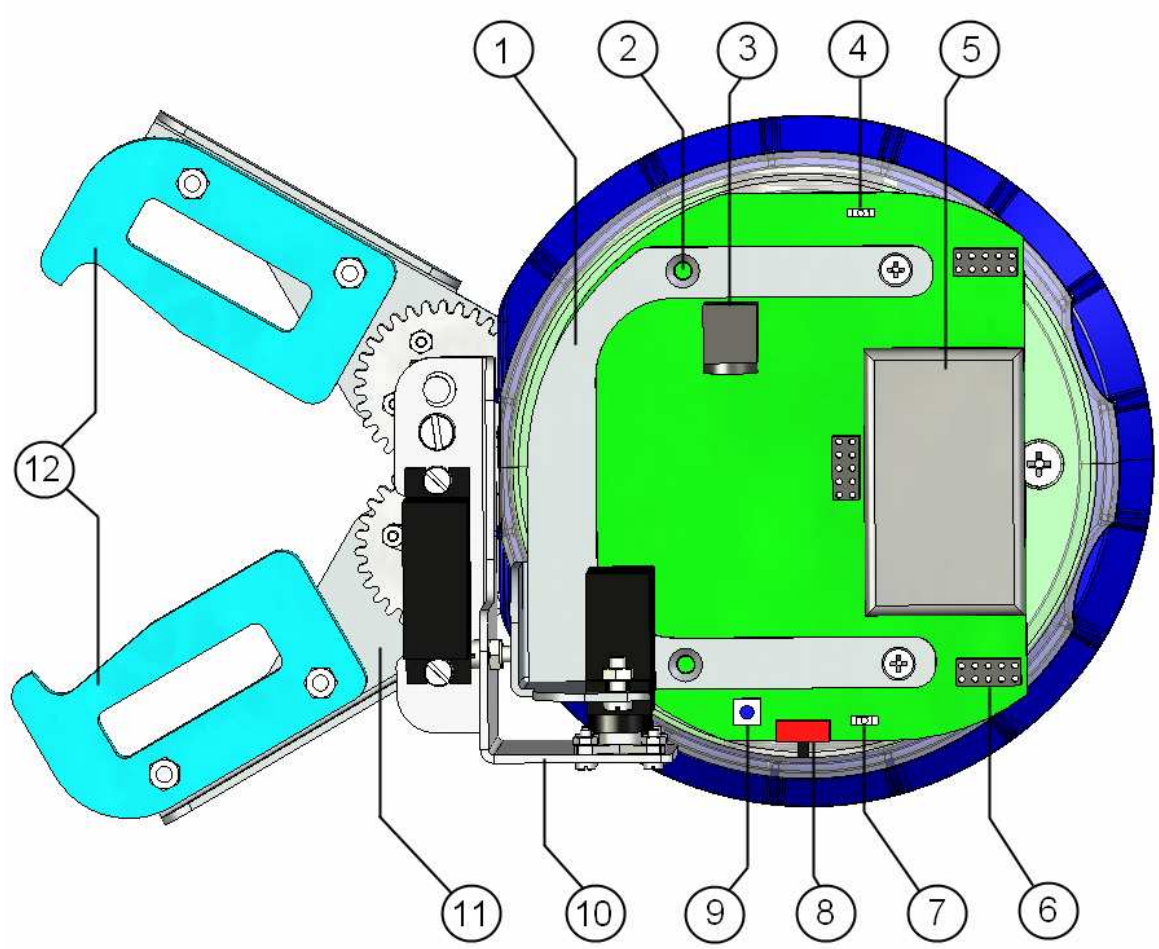


Figure 2.1: Overview of the Gripper turret layout

Make an external inspection of the turret. Note the location of the following parts:

1. Main Support
2. Screws to fix on the robot
3. Power input connector
4. Charge Led
5. Additional battery
6. Extension connectors
7. Power on Led
8. ON/OFF switch
9. Reset button
10. Arm
11. Gripper
12. Foam grip

2.2 Arm

The arm makes the displacement of the gripper possible. The movement of the arm allows to grab object in front of the Robot, and takes it above. This allows the Robot to keep its mobility and all sensors free of obstacle.

The arms is moved by a DC servo motor which is control by a local microcontroller to allows the user to set every position he wants. The movement can be done at the maximum speed, or with a speed regulation to avoid the lurching of the Robot. Different speed can be adjust to match with the application. When you have a heavy load, you must set a slower speed to avoid the lurching of the robot.

The maximum load that can carry the gripper is approximately 30g. The arm can be moved between the position 0 "LOW position"(arm in the lowest position to grab object) and 7000 "TOP position"(arm above the robot against the end stop).

A special mode (active by default) turn off the servo of the arm when it reach the TOP position (between 5600-7000) or the LOW position (0-250). This will increase the battery autonomy when the arm is in stable position.

2.3 Gripper

The gripper is installed at the end of the arm with two foam grip which allows to grab different size and form of object. The user can choose the exact position of the gripper between the maximal open position (5100 = where the gripper will form a angle of 180 degrees) and the closest position (0 = where each foam will touch the other). The maximal size of an object is about 70mm, and the minimal size is about 1mm.

The user can know the exact position of the gripper only when there's no object grabbed. When you grab an object, the servo will try to reach the closed position but cannot reach it (because of the object between the two foam). This will make the tightening.

If you know the size of your object, you can adjust the closed position to the object size. This will allows you to limit the tightening torque, and then limit the current in the motor (to increase the battery autonomy).

A special mode (active by default) turn off the servo of the Gripper when it reach the Open position (between 2300-5100). This will increase the battery autonomy when the Gripper is open.

2.4 Battery

An additional battery is included in the Gripper to increase the autonomy of the Gripper and the K-Junior. This is a LiPol Battery with a capacity of 1350mAh . A external power supply adapter (5V - 1.4A) is deliver with the Deluxe package to charge the gripper Battery. A complete charge take approximately 3 hours. Please be sure that the gripper and the K-Junior are turn off. Otherwise, the charge process can be much longer.

2.5 I²C Address

The K-Junior Robot control the Gripper through an I2C bus. The turret address is defined by 7 bits + 1 bit for the mode selection "write" or "read"

1	0	0	1	0	1	1	R/W

The Gripper I2C address is 0x97 in "read" mode and 0x96 in "write mode".

2.6 I²C Register

Many different register allows the user to configure and use the Gripper

I²C Address	Reading	Writing
0x30	Firmware Version	Do Nothing, read only
0x31 0x32	Read the position of the Arm	Set the Arm position without speed control (0-7000)
0x33 0x34	Read the position of the Gripper	Set the Gripper position without speed control (0-5100)
0x35 0x36	Read the order of the Arm	Set the Arm consign to reach a position with speed control
0x37 0x38	Read the order of the Gripper	Set the Gripper consign to reach a position with speed control
0x39	Read the Arm speed value	Set the Arm speed value for speed control (0-255)
0x3A	Read the Gripper speed value	Set the Gripper speed value for speed control (0-255)
0x3B	Read the Arm turn off flag	Configure the Arm turn off mode (0 = disable, 1 = active)
0x3C	Read the Gripper turn off flag	Configure the Gripper turn off mode (0 = disable, 1 = active)
0x3D 0x3E	Read the Battery voltage	Do Nothing, read only
0x3F	Read the Battery remaining capacity	Do Nothing, read only

All the 16bits register (Position, Consign, Voltage,...) must be read or write through the LSB byte first.

2.6.1 Firmware Version register

This register return the version and revision of the Gripper turret

2.6.2 Position Register (Arm and Gripper)

Reading: This register return the real position of the Arm (or the gripper). The returned value is proportional to the angle of the Arm (or the Gripper). Where 0 is the "LOW position" for the Arm and "closed position" for the Gripper. And the maximum position is 7000 for the Arm and 5100 for the Gripper.

Writing: Setting this register will change the position of the Arm or the Gripper. The servo will immediately move to the position without speed control. Warning, move too fast can make the K-Junior lurching.

2.6.3 Order register (Arm and Gripper)

Reading: This register return the previous order of position value which was set by the user.

Writing: Setting this register will told to the turret to move the Arm or the Gripper to the desired position with a speed control (see Speed Register).

2.6.4 Speed Register

Reading: Read the value of the desired speed of the Arm or the Gripper. 0 is the slowest and 255 the fastest.

Writing: Configure the speed to reach when a move is made with speed control.

2.6.5 Voltage Register

Read the value of the battery voltage (only Deluxe version, return 0 on LE version). This value must be converted as following to obtain the real voltage value:

$$Voltage[V] = \frac{Voltage\,\mathrm{Re}\,gister}{1024} \bullet 5$$

2.6.6 Available Remaining Capacity

Return the Available Remaining Capacity in percent of the battery (only Deluxe version). The calculation of this value is made on the voltage of the battery, so this value is not guarantee to be exact. This is just an evaluation of the battery.

3 Connections

Assembling and disassembling additional turret is a delicate operation. Try to avoid it as much as possible and perform it carefully. Please follow the following instructions to avoid damage to your modules. K-Team can assume no responsibility for any damage caused by improper manipulation

3.1 Assembling

Assembling is quite easy, but it also necessary to perform it carefully:

- First of all choose the parameters of the K-Junior robot and set the switches if necessary.
- Assemble the Gripper turret on the K-Junior directly in two steps: First, place the module on the extension connector checking that all pins are seated correctly. Second, apply force to insert the turret into the extension connector.
- Screw the two screws provided with the gripper on the fixings (see number 8 fig 2.1)
- If you have any other extension, plug it on the Gripper as usual.
- Turn on the Gripper switch and finally turn on the K-Junior

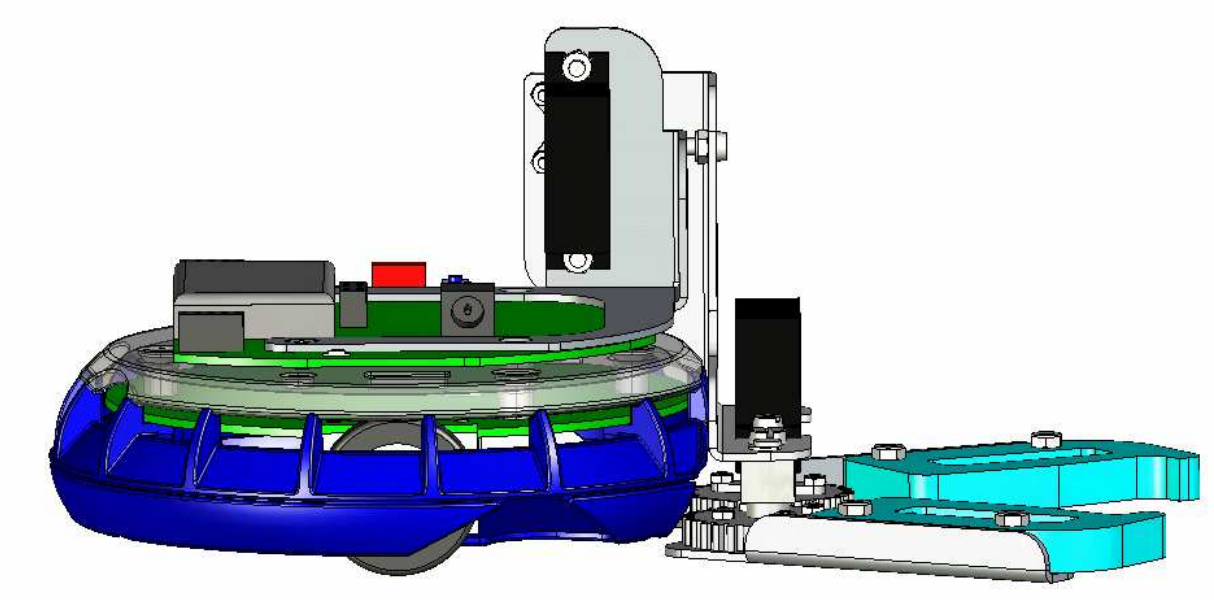


Figure 3.1: Gripper mounted on the K-Junior.

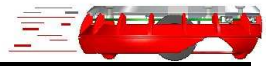
DO NOT try to connect the Gripper on other connector or in another position. This could damage the Gripper turret and the K-Junior.

3.2 Disassembling

For the disassembling, be careful that the Gripper and the Robot must be turned OFF.

- Unscrew the two fixing screws.
- Then take the robot with one hand and the gripper with the other, and pull vertically to avoid to twist the connector pins and broke the connector.

4 Gripper Use



There's two different way to control the Gripper on the K-Junior Robot :

- Programming the K-Junior in C language, using the library of the KJiOs
- Serial remote control (using labview, MatLab, or any other application with a serial protocol).

4.1 Programming in C language

To program the K-Junior with the C language, you must have the CCS compiler. You can buy directly at K-Team a limited version of the compiler to use only with the microcontroller of the K-Junior. Please read first the K-Junior Os manual (http://ftp.k-team.com/K-Junior/KJOSManual.doc) to find all necessary information about the installation of CCS and the use of the KJOs.

- Download now the last version of the KJiOS and the Gripper Library on our website.
- Open the KJOs project with CCS

4.1.1 void HemGripper_Init(void)

Function : Gripper Initialization. This function must be called at the beginning of each program using the Gripper.
Example : HemGripperInit();

4.1.2 char HemGripper_Read_Version(void)

Function : Read the firmware version.
Example : char Version; Version = HemGripperReadVersion();

4.1.3 unsigned int16 HemGripper_Read_Arm_Position(void)

Function : read the position of the Arm (0-7000)
Example : unsigned int16 ArmPosition; ArmPosition = HemGripperReadArmPosition();

4.1.4 unsigned int16 HemGripper_Read_Gripper_Position(void)

Function : read the position of the Gripper (0-5100)
Example : unsigned int16 GripperPosition; GripperPosition = HemGripperReadGripperPosition();

4.1.5 unsigned int16 HemGripper_Read_Arm_Consign(void)

Function : Read the position Consign of the Arm
Example : unsigned int16 ArmConsign; ArmConsign = HemGripperReadArmConsign();

4.1.6 unsigned int16 HemGripper_Read_Gripper_Consign(void)

Function : Read the position Consign of the Gripper
Example : unsigned int16 GripperConsign; GripperConsign = HemGripperReadGripperConsign();

4.1.7 unsigned char HemGripper_Read_Arm_Speed(void)

Function : Read the speed for the Arm movement
Example : unsigned char ArmSpeed; ArmSpeed = HemGripperReadArmSpeed();

4.1.8 unsigned char HemGripper_Read_Gripper_Speed(void)

Function : Read the speed for the Gripper movement
Example : unsigned char GripperSpeed; GripperSpeed = HemGripperReadGripperSpeed();

4.1.9 unsigned char HemGripper_Read_Arm_Disable(void)

Function : Read the flag which indicate if the Arm servo must be turn off or not when it reached a stable position
Example : unsigned char ArmDisableFlag; ArmDisableFlag = HemGripperReadArmDisable();

4.1.10 unsigned char HemGripper_Read_Gripper_Disable(void)

Function : Read the flag which indicate if the Gripper servo must be turn off or not when it reached a stable position
Example : unsigned char GripperDisableFlag; GripperDisableFlag = HemGripperReadGripperDisable();

4.1.11 unsigned int16 HemGripper_Read_Battery_Voltage(void)

Function : Read the value of the battery voltage
Example : unsigned int16 BatteryVoltage BatteryVoltage = HemGripperReadBatteryVoltage();

4.1.12 unsigned char HemGripper_Read_Battery_Capacity(void)

Function : Read the value of the available remaining capacity of the battery
Example : unsigned char Capacity Capacity = HemGripperReadBatteryCapacity();

4.1.13 void HemGripper_Set_Arm_Position(unsigned int16 value)

Function : Set the new position of the arm. The servo will reach this position as fast as possible
Example : HemGripperSetArmPosition(3500);

4.1.14 void HemGripper_Set_Gripper_Position(unsigned int16 value)

Function : Set the new position of the gripper. The servo will reach this position as fast as possible

Example : HemGripperSetGripperPosition(100);

4.1.15 void HemGripper_Set_Arm_Consign(unsigned int16 value)

Function : Set the consign of the arm. The Arm will reach this position with the defined speed

Example : HemGripperSetArmConsign(2000);

4.1.16 void HemGripper_Set_Gripper_Consign(unsigned int16 value)

Function : Set the consign of the Gripper. The Gripper will reach this position with the defined speed

Example : HemGripperSetGripperConsign(4000);

4.1.17 void HemGripper_Set_Arm_Speed(unsigned char value)

Function : Set the speed consign for the movement of the arm.

Example : HemGripperSetArmSpeed(100);

4.1.18 void HemGripper_Set_Gripper_Speed(unsigned char value)

Function : Set the speed consign for the movement of the Gripper.

Example : HemGripperSetGripperSpeed(100);

4.1.19 void HemGripper_Arm_Disable(unsigned char value)

Function : Configure if the Arm must turn off its motors once a stable position is reached.

Example : HemGripperArmDisable(1);

4.1.20 void HemGripper_Gripper_Disable(unsigned char value)

Function : Configure if the Gripper must turn off its motors once a stable position is reached.

Example : HemGripperGripperDisable(0);

Remember : to use these functions, you must first include the Gripper library in the beginning of your source code: #include "HemGripper.h" (the file must be in the same directory of course)

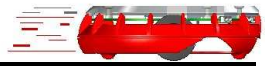
4.2 Serial Remote Control

From the K-Junior firmware version A-01, you can access to the I2C modules through the serial protocol. If you have an older version of firmware, you can update it by downloading the last version on our website. Please have a look at the last K-Junior User Manual to learn more about the serial protocol. Example :

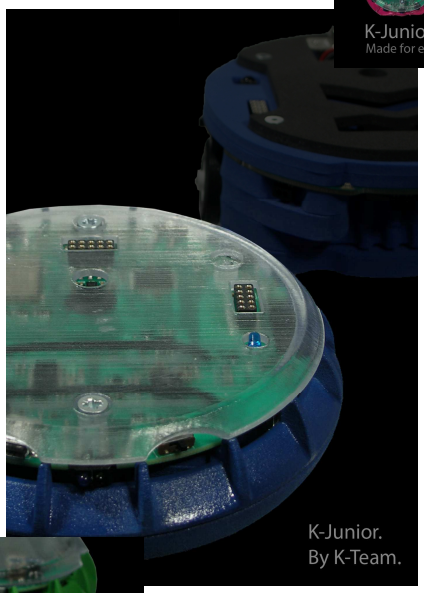
- R,97,00 : Read the Gripper Firmware version

This property can be use with any software which have a serial communication protocol, like sysquake, Matlab,...

A Technical Specification



- Weight : 200 [g]
- Power Supply : 5 [V]
- Maximum payload: 30[g]
- Minimum object size: 1[mm]
- Maximum object size: 70[mm]
- Autonomy: up to 3 hours



K-Team
MOBILE ROBOTICS

K-TEAM S.A.
RUE GALILEE 9
1400 YVERDON-LES-BAINS
SWITZERLAND

A DIVISION OF
KT & GT
Group